Autonomous Swarms of High Speed Maneuvering Surface Vessels

Joshua Vander Hook[1], William Seto[1], Viet Nguyen[1], Zaki Hasnain[1], Liam Gallagher[1], Tyler Halpin-Chan[2], Varun Varahamurthy[2], and Moises Angulo[2]

[1] NASA Jet Propulsion Laboratory, Maritime and Multi-Agent Autonomy Group, Calfornia Institute of Technology, Pasadena, CA, USA, {hook,seto}@jpl.nasa.gov
[2] Surface Targets Engineering Branch 53150ME, Building 466, CBC Port Hueneme, CA, USA, {tyler.halpin-chan,varun.varahamurthy,moises.angulo1}@navy.mil

ABSTRACT

We review efforts by NAVAIR and NASA Jet Propulsion Laboratory to automate the operation of the largest fleet of autonomous maritime vehicles. The vehicles are intended for large-scale demonstrations of US Navy systems and tactics to counter asymmetric naval threats. This review covers a distributed architecture for human-in-the-loop control of several autonomous high speed boats. Hazard avoidance and formation control was verified using real-world vehicles.

1. INTRODUCTION AND OBJECTIVES

The Seaborne Targets Team (STT) as part of Threat/Target Systems Division of NAVAIR has been converting boats into remotely controlled targets to support US Navy operations testing for the past 19 years. As part of their ongoing development in Port Hueneme, California, STT maintains the largest fleet of remote-controlled aquatic vehicles in the world. The vehicles are used to test surface operations, tactics, and weapons for the US Navy. They are also used as research vessels for autonomous maritime development by the Office of Naval Research, NASA Jet Propulsion Laboratory, and many others. To accommodate new and developing threat models, to reduce operator requirements, and modernize for research, STT has partnered with NASA Jet Propulsion Laboratory to build autonomous and semi-autonomous capabilities into the vessels in a low-cost, off-the-shelf platform. A primary goal is to increase the number of vehicles simultaneously commanded by a single driver, with the eventual goal of fielding a fully-autonomous "red team" for test and evaluation of tactics and systems against a fast-moving, asymmetric opponent. Development for this effort is focused on the High Speed Maneuvering Surface Targets (HSMST — Figure 1a).

In this report, we present progress on the partnership between STT and the NASA Jet Propulsion Laboratory to bring swarm-based motion planning to the fleet. The architecture has numerous trade-offs that emphasize human-in-the-loop operation, while easing the operator burden and allowing command of several vehicles simultaneously. We also discuss the future autonomy road map, and the ongoing research efforts.

This paper is intended to remain high level due to size constraints and is organized as follows. First, we overview the existing systems employed by STT in Section 2. The system has been in operation for over 19 years, and has a long legacy which is being modernized as part of this effort. In section 3 we discuss the capabilities being built into the system, and the design requirements for human-in-the-loop operations. Beginning in Section 3.1 we discuss our implementation of the capabilities and present field-testing results from 2017 and 2018 in Section 4. In the end, (Section 5) we discuss preliminary designs of more advanced capabilities, which are currently undergoing sea trials and active research. As a distributed robotic system, the network of boats and human operators is not reliant on any particular centralized computing resource.

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Government sponsorship acknowledged. Copyright © 2018, California Institute of Technology.

(a)
Figure 1: The High-Speed Maneuvering Surface Target (HSMST).

2. ARCHITECTURE OVERVIEW

The boat under test is the High Speed Maneuvering Surface Target (HSMST). The HSMST is an 8-meter-long commercial rigid hull inflatable boat (RHIB). It has two outboard engines that enable it to achieve high speeds in fair and moderate seas. The HSMST is used in a variety of systems testing or training scenarios to replicate various high speed surface threats. The HSMST has a local network of controlling nodes, interconnected by a Controller Area Network (CAN) bus. The network of all nodes on each boat, all boats, and all operator stations is called the SeaCAN network. A system diagram for the local SeaCAN network on each agent is shown in Figure 2. Each “node” or single-board computer is built on inexpensive COTS platforms and has a single requirement. For example, there are nodes to servo the vehicles controls (e.g. rudders and throttles), a node to run the engine interfaces, a node for instrumentation, one for SeaCAN communications, and a removable node for autonomous control (Autonomy). The Autonomy node can therefore be removed and transferred to a different system type. The modular design has been verified on various types of vehicles from personal watercraft to ships.

The remote control system provides (via the onboard nodes) for throttle, rudder, engine, and auxiliary devices control (if present). It also provides operator with the following information via telemetry: speed, engine RPM, GPS location, and, heading. The command control and telemetry exchange is accomplished via Radio Frequency link. The Command interface is called PCCU (Figure 3). At minimum, it allows joystick control of any vehicle remotely. When an autonomy node is present on the vehicle, it also supports heading / course-keeping, drag-and-drop waypoint control for one or more vehicles, transits in formation, and several demonstration-specific commands like following arcs or weaving paths.

STT has gathered enough HSMST performance data to provide high-fidelity model of dynamic response to control inputs. This model is used by autonomy to conduct model-predictive control as discussed in Section 3.1.

Each HSMST can be remote operated, directed by onboard autonomy, or controlled by an onboard safety driver. Autonomous control is easily overridden by the onboard safety operator in the event of unsafe behavior. Safety drivers are always on board except for the most aggressive test regimes which might endanger operator safety because of rough seas or live fire. Drivers are also not used when a human operator cannot react quickly enough because the boats are moving too fast, or are too close.

Since the boats operate over large areas (many kilometers), the radio data rate is necessarily low. The SeaCAN communication protocol is on a mesh network. As the number of boats scales upward, the amount of information each boat can transmit drops, simply by virtue of congestion. Dealing with communication congestion in large vehicle swarms is a very challenging issue. As a working example, consider a formation transit. Each vehicle might receive a command specifying the boats that are in the formation, their points in the formation, and the current heading and speed of the formation. Given N vehicles, it would therefore have to contain $\mathcal{O}(N)$ bytes,

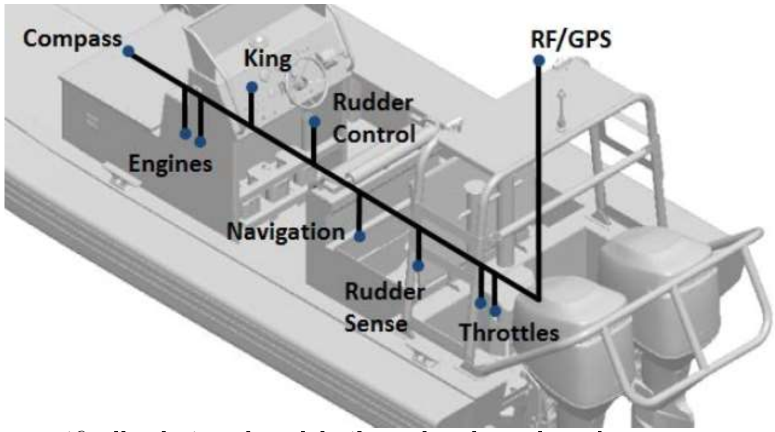


Figure 2: SeaCAN is specifically designed and built to be the onboard remote control system for sea-borne targets. It uses the CAN bus to send a set of standardized messages between microcontrollers called nodes. Each node has a specific function and may be coupled with a sensor or actuator to control and monitor the functions for the remote-controlled sea-borne target. The modular design provides the flexibility to be used in a variety of sea-borne platforms from personal watercraft to ships.

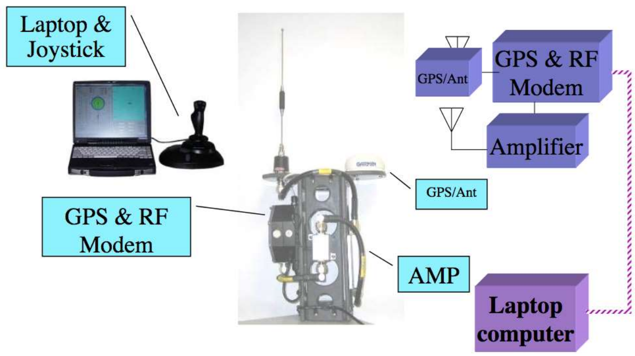


Figure 3: A block diagram of the PCCU (Portable Command and Control Unit). The unit connects to the SeaCAN network to transmit commands to each vehicle on the network, including remote-control or commands to Autonomy nodes.

and the total communication would be $\mathcal{O}(N^2)$. Instead, SeaCAN commands are structured so that each one only contains the information relevant to the commanded vehicle. In this way, the communication scales as $\mathcal{O}(N)$.

The only explicit exchange of information between the boats is a shared world model. Each boat broadcasts its position during its time slot, as measured by onboard GPS/INS. Adapting the boats to use sensing and estimation of the other boats' positions is outside the scope of this effort, but is being explored for other programs.

This introduces a difficulty in designing Autonomy, however, since now vehicles cannot explicitly exchange information with each other. Therefore, all coordination must be *implicit*, i.e., control and decision processes are necessarily distributed. This distributed autonomous system must still be operationally intuitive, fluid, and responsive to humans providing input to the swarm.

3. AUTONOMY REQUIREMENTS AND DESIGN

When Autonomy is given control of the vehicle, it is responsible for the safety of the vessel and operator (if present), as well as responding to operational commands. The set of operational commands (or "behaviors") are discussed in this section, along with the high-level concepts of their implementation. The Autonomy node designed for STT at NASA JPL was built on top of the CARACaS autonomy framework for distributed robotic systems.[1–3] The autonomy "stack" is as shown in Figure 4. The motivation, requirements, and implementation are covered in this section.

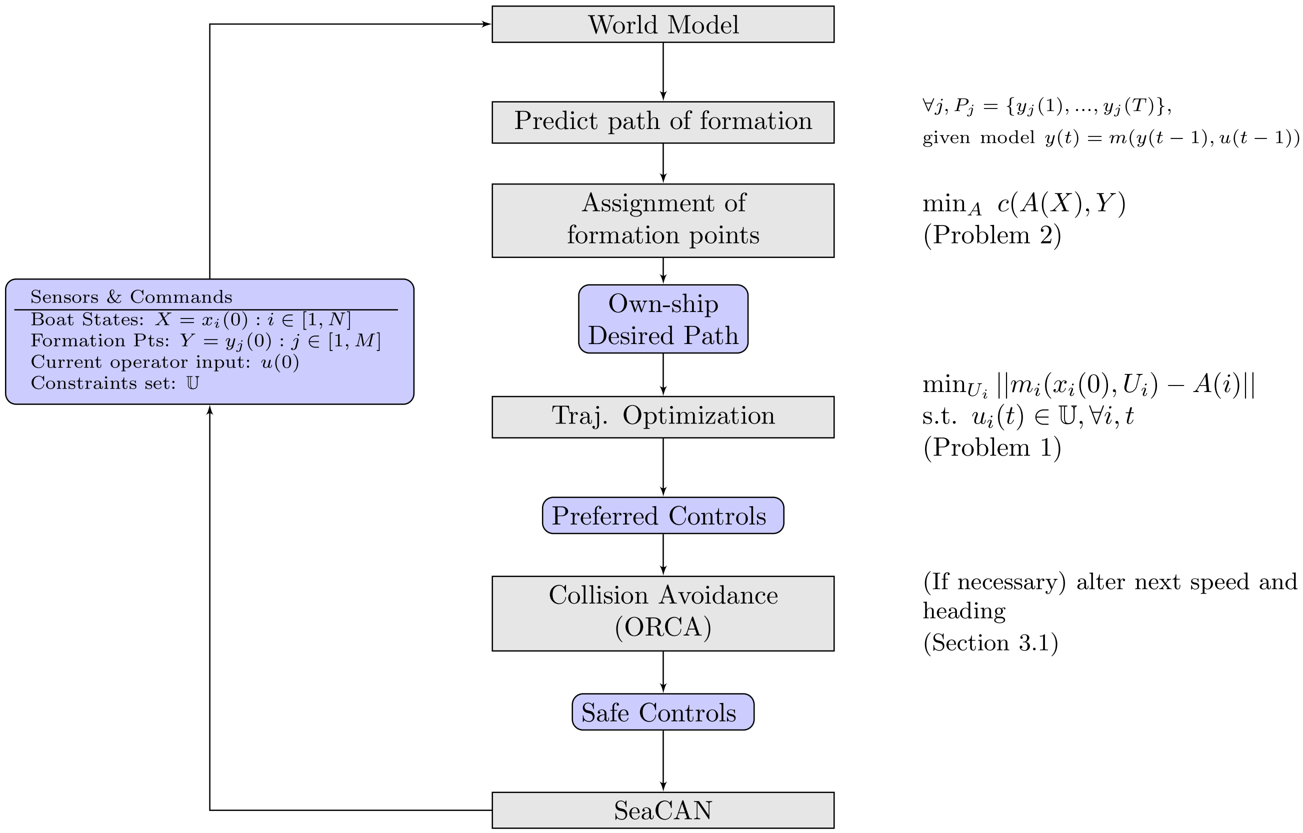


Figure 4: Autonomy node control loop diagram. The SeaCAN bus connects autonomy to all other nodes and vehicles. This architecture reflects the 2017-2018 test events.

Autonomous HSMSTs can be issued commands individually or as a group. Groups are how autonomy is aware of which vessels it should cooperate with during commands. For example, autonomy on one vehicle is aware that it should be in formation with other vehicles because it has received a formation transit command and knows which vehicles are in its group. As discussed, to save bandwidth it is not aware of other groups' commands, or the details of the commands from vehicles in its own groups. For example, during changes in formation, it is possible for two boats to have to swap positions. However, each boat is only given the set of parameters (e.g., formation positions) relevant for *its own* operation. To be effective with limited bandwidth and almost no ability to exchange information, autonomy is built to infer the intended actions of the other vessels.

3.1 Autonomy Capability Overview

Remote Operation: Each HSMST can be remote operated by an operator on the network. The PCCU allows joystick input, which is passed to the onboard navigation node. Autonomy is disabled during this mode, and any autonomy commands are ignored.

Waypoint Navigation and Velocity Keeping: Autonomy is responsible for setting a desired speed and heading, and lower-level, pre-existing PID controllers are provided on each vessel via the navigation node. This is to allow sufficient abstraction so that autonomy nodes can be moved between vehicles easily. Waypoint navigation consists simply of a line-following algorithm implemented as a feed-forward proportional controller. Velocity control is passed through to the vehicle's PID (throttle and rudder control). However simple, during these modes, the autonomy node must take into account world state to keep the vehicle safe from collisions by steering around them.

Safety of Navigation: To accommodate safe operations the US Navy has set up a framework for determining safe operating speeds, minimum safe separation between vessels, and safe command sequences during testing.

Several factors play into the choice of safe speeds. The most relevant factors are: proximity of operations, presence of humans, and sea state. In the future, it will be automatically determined by using a sea-state aware path planner, as discussed in Section 5. However, speed constraints are currently specified by human operators as part of command messages.

Whatever the safe speed, collisions during autonomous operations are a serious risk whenever human operators are onboard. Each vessel has a safe separation distance. As with safe speeds, standoff distances take into account several factors, such as waves, winds displacement, speeds of operations, and types of maneuvers. At full speeds and in close proximity improper maneuvers can cause a collision in fractions of a second. Thus, override by a human operator is infeasible, and autonomy is expected to intervene. Onboard autonomy has two ways of intervening to avoid collisions when human operators are present.

The simplest of safe-keeping methods is to stop all vessels that get too close. In the event that any two boats violate a safety standoff distance, autonomy is designed to stop the vessel. However, if the vehicles detect an impending collision, but have not yet reached the standoff distance, they can recover by coordinating their maneuvers to avoid each other. Several methods for using controls or path planning to avoid collisions are present in literature. A few of the techniques are velocity obstacle and its variations,[4–7] model predictive control,[8,9] artificial potential fields,[10,11] and roadmap based approaches such as visibility graphs,[12,13] probabilistic roadmaps,[14,15] and Voronoi Diagrams.[16,17] For this work, we chose Velocity-Obstacle-based collision avoidance, in particular Optimal Reciprocal Collision Avoidance (ORCA).[18]

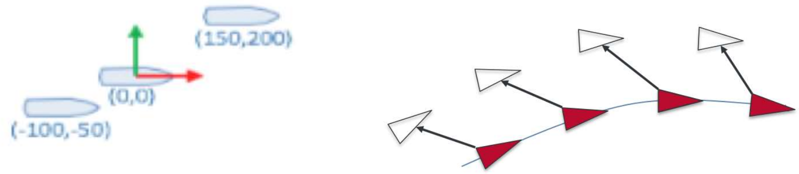
NASA JPL has a long history of projects using velocity obstacle-based collision avoidance.[19] The collective decision making implicit in ORCA has several tangible advantages on our formation control design. In traditional, single-agent VO, a boat in the middle of a tight formation would initiate a hazard avoidance maneuver, but would be constrained by its neighbors, resulting in a solution that requires slowing down. However, using ORCA, all boats take partial actions to allow the middle boat to avoid collision, i.e., splitting the formation.The most important part of OCRA, relevant to this effort, is that each boat can "share" the responsibility of avoiding collisions. As a result, the boats can operate safely at higher speeds since control effort for keeping vessels safe is spread among all the vehicles and "pile-ups" of vehicles are avoided (figuratively so far, but possibly literally).

Formation Control: The control of many vehicles by a single operator was accomplished by human-in-the-loop formation control. A human operator controls a virtual boat, which appears in the situational awareness picture of each autonomous vessel (i.e., it's location and velocity, is broadcast over the network). The boats then "form-up" with the virtual boat. Note, a real, human operated vehicle can be used without significant change. The virtual boat is the *leader* in what follows, and can be steered through any maneuver. The result is a swarm of vehicles maneuvering together in formation.

A formation can be preset, so that each vessel knows its assigned position relative to the leader, or adaptive, so that vessels must coordinate to converge into the desired formation. Preset formations are specified as potentially time-varying offsets $o_i(t)$ for the i^{th} boat, from the virtual leader position $x_v(t)$. Each boat receives a pair denoting the offset specified in a coordinate frame centered on the virtual leader and aligned with its direction of travel (Figure 5a). By varying the offset, the boats can undergo a formation transition. In field experiments, we tested all of the formations and transitions between formations shown in the transition diagram in Figure 6.

The virtual leader is "driven" by a human operator, and its position follows a kinematic model based on the joystick commands of the human operator. To converge to the desired formation point around the virtual leader, each boat solves a trajectory optimization problem as follows. Let the virtual boat's position be $x_v(t)$ and virtual boat's commanded speed and heading be $u_v(t)$ at the current time t. Using the model by which the virtual leader's position evolves, $x_v(t+1) = m_v(x_v(t), u_v(t))$, each boat knows its desired trajectory for the next T seconds by $X_i^\star = m_v(x_v(t-1), u_v(t-1)) + o_i(t)$ for t up to T (Figure 5b).

Problem 1 (Formation Point Convergence). *Choose a control sequence for the next T steps, $U_i = u_i(t), \cdots, u_i(t+T)$, to bring the vehicle's predicted position $X_i = m_i(x_i(t), U_i)$ to the path of the moving formation point, $X_i^\star = m_v(x_v(t-1), u_v(t-1)) + o_i(t)$ (Figure 5b). The vector $\mathbb{U}$ represents the bounds on the control allowable, such as maximum safe speed and turn rates. The constraints are given as part of the command sequence, and not known a priori.*



(a) (b)

Figure 5: (a) The offsets for each boat from the leader boat (at the origin) are transmitted as part of the command from SeaCAN. (b) Problem 1 for a single vehicle. A real (or simulated) leader (Red) is moving on a trajectory determined by a remote operator. The formation-keeping task is to control each HSMST so that they stick to their offset (White), as defined by the leader position.

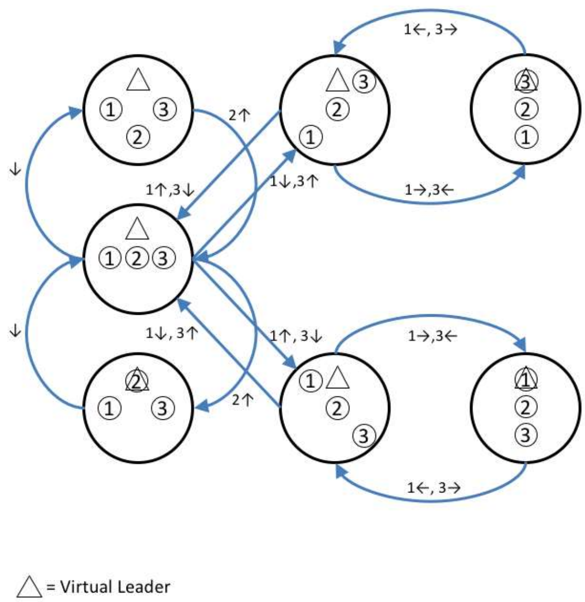


Figure 6: Formation transition diagram showing the changes in formation commands verified during field trials.

Our preferred control method to solve Problem 1 is a forward-shooting trajectory optimization algorithm. Our control model for the vessel is a unicycle model. We use model predictive control[20] to lead the vehicle's PID controller, to which we do not have direct access. Actuation is not instantaneous, and so we have a simple model that predicts the convergence of the vehicle to the desired heading and speed.

Adaptive Formations: Adaptive formations allow the boats to their position based factors other than operator input. There are two kinds of adaptive formations. In the first, the formation points are specified fully, but not which agent should occupy which position. We define these as *anonymous formations*. The second kind is *shape formations*, in which a general shape is defined, but formation points must be calculated by all boats simultaneously. A key feature of our distributed autonomy system is: Given knowledge of each agent's position and the set of possible positions, each boat will converge to the same solution so that they agree upon the assignment without exchange of information.

Three key capabilities enabled by re-assigning formation points are *arbitrary startup*, *dynamic groups*, and *reshuffle*. Arbitrary startup allows a formation to quickly re-form when starting (or continuing) a formation transit (See Figure 9). Dynamic groups allow the formation to accommodate vessels joining or leaving formations safely and with minimal control effort. Reshuffle allows the formation to accommodate sharp turns without endangering vehicles by swapping formation points. For example, a line-abreast quickly reversing its course.

We deal with *anonymous formations* as follows. First, each boat receives a command specifying an anonymous

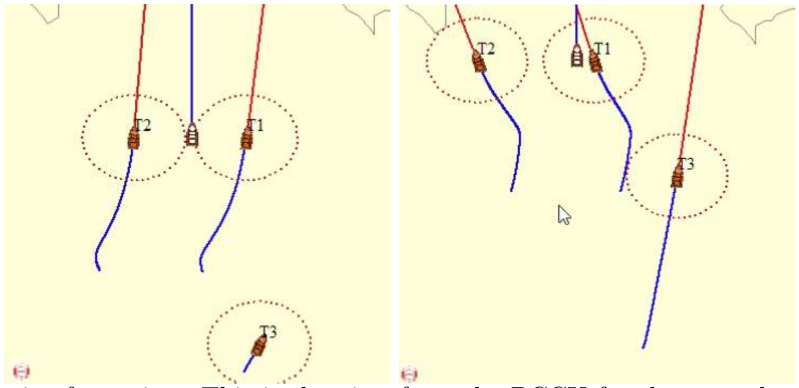
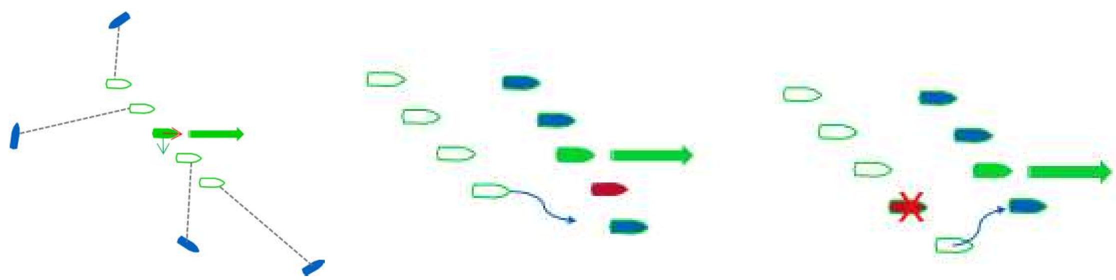


Figure 7: A live adaptive formation. This is the view from the PCCU for the control station during the 2017 tests.



(a) (b)

Figure 8: (a) From arbitrary position, the boats use a deterministic assignment algorithm to ensure agreement. (b) Adapting to a new boat (red) joining or leaving a formation. The leader is in Green and direction of travel shown.

formation. The types of anonymous formations are limited to a specific, parameterizable type ("Wedge", "Line", etc). The parameters of the formation are usually spacing between boats, but may also include rotations or affine transformations. At any rate, by specifying formations in this way, a small number of bytes broadcast to all boats allows them to avoid any ambiguity in the steady-state desired locations of the boats. The boats then solve the following problem to determine their own desired formation location.

Problem 2 (Formation Assignment). *N vehicles at points $X = x_1, \cdots, x_N$ are given a set of formation points $Y = y_j, \cdots, y_M$ for $M \geq N$. Each point y is moving with the same course and speed u_v. All agents must choose a mapping $A : X \to Y$ such that the total intercept cost $c(A(X), Y)$ is minimized.*

Depending on the operational regime, $c(\cdot)$, is usually either maximum, or sum of, the cost to intercept the moving formation point on a straight-line, maximum-speed trajectory. Both versions are easily minimized given elementary methods in literature, e.g., the Hungarian Assignment algorithm.[21] The assignment regimes can be free (boats take any point in the formation) or rank-based, such that N boats always occupy the first N points in the formation.

When the formation points Y are not explicitly known, they must be derived by all boats simultaneously, then assigned. In Figure 7, the boats must form a line, moving abreast with the motion of the virtual leader. The line of the boats must be centered on the virtual leader, and may contain an arbitrary number of vessels. The formation points are constrained to be a fixed distance apart, and the center of the line is set to coincide with the virtual boat's position. Other, more complicated adaptive formations are the subject of future work.

4. FIELD EXPERIMENTS

From Summer of 2017 through March, 2018 the motion planning pipeline was verified using up to four HSMSTs and human operators in Oxnard, California, USA. First, hazard avoidance was verified as follows. We defined four scenarios for two-boat interaction. Head-on collisions, crossing left and right, and overtaking. We also tested

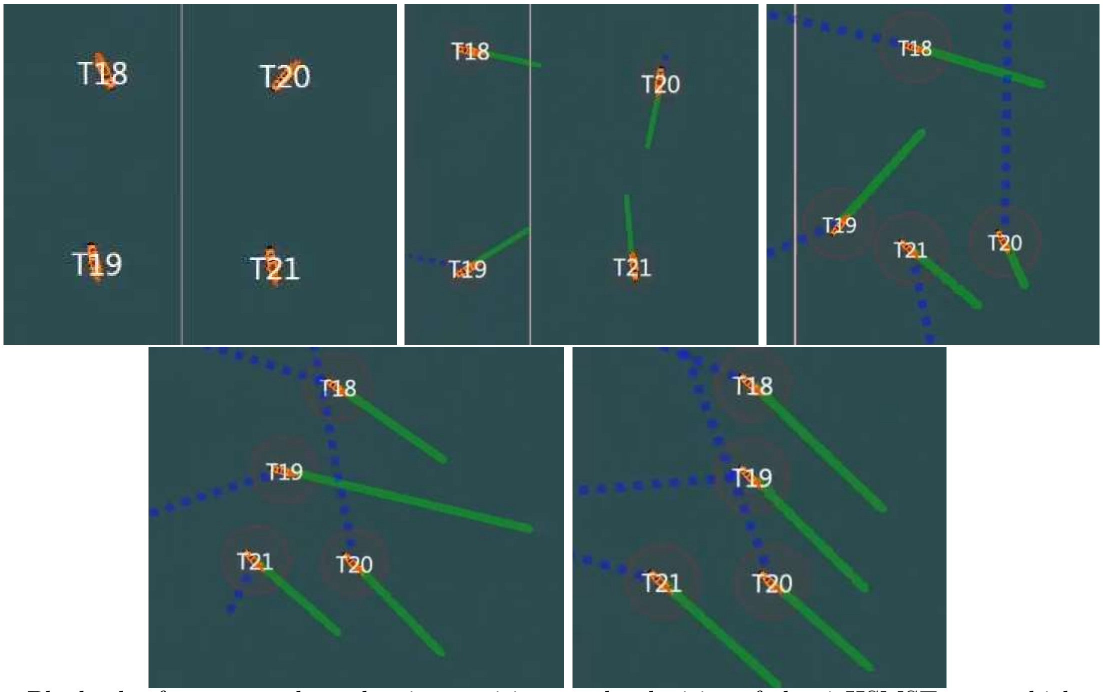


Figure 9: Playback of on-water data showing positions and velocities of the 4 HSMST test vehicles. This is the view from the PCCU control station during the 2018 tests. The green lines show velocity vectors of length proportional to velocity. The red circles around each vehicle is the standoff distance (overlapping circles is a failure). The boats were travelling at up to 25 knots during these tests.

up to four vehicles in a “pile-up” where they were all ordered to pass through the same point. These tests were used to validate hazard avoidance and re-planning before beginning any formation transit tests. Due to space, we omit the explicit details of these tests.

Validation criteria were a lack of safety buffer violation. Qualitatively, the transitions were evaluated for intuitive results. The field test were conducted over a series of 10 months and produced no safety-related shutdowns. The ORCA algorithm in particular was successful at preventing safety buffer violations and produced intuitive results. However, metrics are required to do more than qualitative comparison of path planning and hazard avoidance techniques in this domain. Ongoing validation efforts using time-to-convergence metrics are planned for Summer 2018.

Formation Transits using the formations in Figure 6 were verified during straightaway and turning maneuvers. Of particular importance is the smooth transition between changes in formations during command changes, and all transitions in the figure were also verified multiple times. We have already presented a subset of the results in Figures 9 and 7. Figure 10 shows another example of a long transit of three vehicles under a single operator control.

5. CONCLUSION AND FUTURE CAPABILITIES

In this paper we have discussed a fully-tested distributed architecture for autonomous surface vessels. The limited bandwidth available for communication necessitates that each vessel uses mostly onboard information to cooperate. Cooperative behaviors were designed and field tested using a live system, verifying the design against operational requirements.

Our ongoing development has targeted three main capabilities. They are: Sea state aware guidance, tight formations, and larger swarms. We briefly discuss our current directions for each of these.

Currently, human operators set the maximum safe speed based on observed sea state, and the trajectory optimization and control algorithms operate agnostic of sea state. Hull slam and planing effects can damage

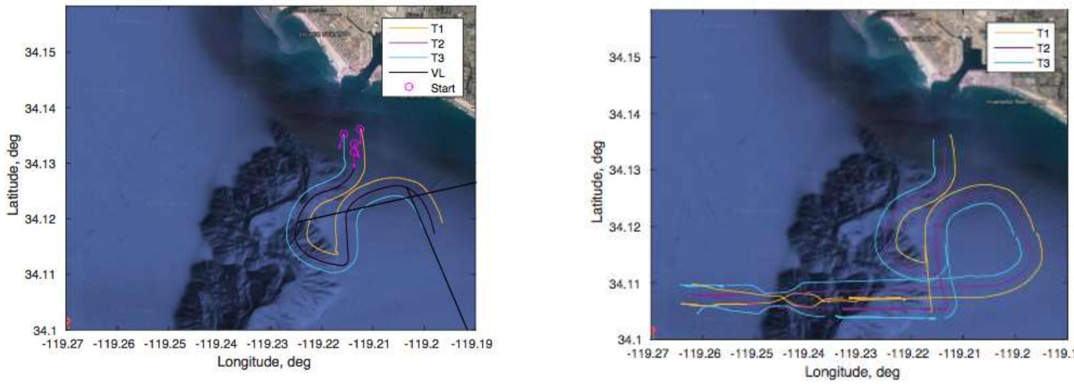


Figure 10: Two examples of long, steady-state formation transits under remote control. (North is up). Note, in the second figure the formation begins in the upper right, transitions along the paths shown to the bottom left, then turns 180 degrees abruptly (reshuffling assignments) and then proceeds east through a pair of formation transitions.

a vehicle or make it uncontrollable by conventional algorithms. Prior work on sea-state-aware controllers can seamlessly integrate into Velocity Obstacle planners.[22] Inclusion of sea-state awareness would allow vehicles to simultaneously choose collision-free velocities which are also safe for the vehicles. However, planing effects such as those experienced by fast-moving vehicles require special care, since dynamics and control become non-linear in these regimes.

Reducing the offsets between vehicles during transit poses a significant technical challenge. The close proximity leaves less room for control errors and requires much tighter synchronization of all the vehicles' motion plans. Without explicit communication, it might be required that each vehicle more accurately predict the motion of the others' to form a safe motion plan such as.[23]

The architecture we have discussed scales easily with swarm size. We have verified ORCA for 40 agents, and found no significant slowdown (nearly linear scaling). For larger swarms, the main bottleneck in this framework is the assignment step in Problem 2. Addressing these problems without requiring additional communication or exchange of information will remain a key challenge.

REFERENCES

[1] Huntsberger, T., Aghazarian, H., Castano, A., Woodward, G., Padgett, C., Gaines, D., and Buzzell, C., "Intelligent autonomy for unmanned sea surface and underwater vehicles," *AUVSI Unmanned Systems North America* (2008).

[2] Huntsberger, T. and Woodward, G., "Intelligent autonomy for unmanned surface and underwater vehicles," in [*OCEANS 2011*], 1–10, IEEE (2011).

[3] Woodward, G., Gierach, M., Schaffer, S., Chu, S., Estlin, T., Castano, R., and Li, Z., "Adaptive auv in-situ sensing system," in [*OCEANS–Anchorage, 2017*], 1–4, IEEE (2017).

[4] Fiorini, P. and Shiller, Z., "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research* **17**(7), 760–772 (1998).

[5] Van den Berg, J., Lin, M., and Manocha, D., "Reciprocal velocity obstacles for real-time multi-agent navigation," in [*2008 IEEE International Conference on Robotics and Automation*], 1928–1935, IEEE (2008).

[6] Wilkie, D., Van Den Berg, J., and Manocha, D., "Generalized velocity obstacles," in [*2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*], 5573–5578, IEEE (2009).

[7] Snape, J., Van Den Berg, J., Guy, S. J., and Manocha, D., "The hybrid reciprocal velocity obstacle," *IEEE Transactions on Robotics* **27**(4), 696–706 (2011).

[8] Park, J., Kim, D., Yoon, Y., Kim, H., and Yi, K., "Obstacle avoidance of autonomous vehicles based on model predictive control," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* **223**(12), 1499–1516 (2009).

[9] Ji, J., Khajepour, A., Melek, W. W., and Huang, Y., "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Transactions on Vehicular Technology* **66**(2), 952–964 (2017).

[10] Ge, S. S. and Cui, Y. J., "Dynamic motion planning for mobile robots using potential field method," *Autonomous robots* **13**(3), 207–222 (2002).

[11] Warren, C. W., "Multiple robot path coordination using artificial potential fields," in [*Proceedings., IEEE International Conference on Robotics and Automation*], 500–505, IEEE (1990).

[12] Alt, H. and Welzl, E., "Visibility graphs and obstacle-avoiding shortest paths," *Zeitschrift für Operations-Research* **32**(3-4), 145–164 (1988).

[13] Kunchev, V., Jain, L., Ivancevic, V., and Finn, A., "Path planning and obstacle avoidance for autonomous mobile robots: A review," in [*International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*], 537–544, Springer (2006).

[14] Kavraki, L., Svestka, P., and Overmars, M. H., [*Probabilistic roadmaps for path planning in high-dimensional configuration spaces*], vol. 1994, Unknown Publisher (1994).

[15] Hsu, D., Latombe, J.-C., and Kurniawati, H., "On the probabilistic foundations of probabilistic roadmap planning," *The International Journal of Robotics Research* **25**(7), 627–643 (2006).

[16] Zhou, D., Wang, Z., Bandyopadhyay, S., and Schwager, M., "Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells," *IEEE Robotics and Automation Letters* **2**(2), 1047–1054 (2017).

[17] Breitenmoser, A., Schwager, M., Metzger, J.-C., Siegwart, R., and Rus, D., "Voronoi coverage of non-convex environments with a group of networked robots," in [*2010 IEEE International Conference on Robotics and Automation*], 4982–4989, IEEE (2010).

[18] Van Den Berg, J., Guy, S. J., Lin, M., and Manocha, D., "Reciprocal n-body collision avoidance," in [*Robotics research*], 3–19, Springer (2011).

[19] Kuwata, Y., Wolf, M. T., Zarzhitsky, D., and Huntsberger, T. L., "Safe maritime autonomous navigation with colregs, using velocity obstacles," *IEEE Journal of Oceanic Engineering* **39**(1), 110–119 (2014).

[20] Camacho, E. F. and Alba, C. B., [*Model predictive control*], Springer Science & Business Media (2013).

[21] Phillips, D. T. and Garcia-Diaz, A., [*Fundamentals of network analysis*], Prentice Hall (1981).

[22] Ono, M., Quadrelli, M., and Huntsberger, T. L., "Safe maritime autonomous path planning in a high sea state," in [*American Control Conference (ACC), 2014*], 4727–4734, IEEE (2014).

[23] Alonso-Mora, J., Breitenmoser, A., Rufli, M., Beardsley, P., and Siegwart, R., "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in [*Distributed Autonomous Robotic Systems*], 203–216, Springer (2013).